NSCCS User Guide

NSCCS User Guide

Introduction

This introductory guide provides users with the information they will need to access and use the computing resources provided by the EPSRC UK National Service for Computational Chemistry Software (NSCCS). We aim to keep this information up to date but users should refer to the NSCCS web site (http://www.nsccs.ac.uk) for the latest news and service information.

Disclaimer

This user guide is provided for information purposes only. Although thorough checks have been carried out on the contents of the pages, there could still be some errors remaining. The NSCCS do not accept responsibility for any errors caused due to reference to any of the pages from this user guide, and it is also not responsible for the content of external internet sites quoted and does not endorse any of the material on these links.

EPSRC Pioneering research and skills

NSCCS

Imperial College London

with our partner at Science & Technology Facilities Council e-Science

CONTENTS

1 Registration

1.1 Getting a Userid

When a project has been approved, all group member(s) or collaborator(s) specified by the Principal Investigator (PI) on the application form will be allocated an account on the NSCCS machine, unless they already have a valid Rutherford Appleton Laboratory (RAL) userid. New users will have a special online registration web link emailed to them by the Service Manager and they will be asked to sign a Declaration Form agreeing to the terms and conditions for use of our software and the STFC RAL data protection act. The '*Terms and Conditions of Use*' can be found on our website at:

http://www.nsccs.ac.uk/termsofuse.php

Once they have signed the forms electronically, their RAL userid and password will be sent through the post.

Any group member or collaborator who was not specified in the original application may be added at a later date. To do this, the PI should send an email to the Service Manager with the name and email address of the user to be added.

If a user has forgotten his/her password, they should contact NSCCS Support by email (nsccs.support@stfc.ac.uk).

1.2 How to Change a Password

Users are advised to change their passwords as soon as they log in to the NSCCS machine (see section 2). This can be done by typing the following command at the Unix prompt:

```
passwd
```

You will be prompted for your current password (Old password) and then asked for a new password which you will need to repeat.

2 Accessing the machines

2.1 Hardware

The NSCCS hardware is based and managed at the Rutherford Appleton Laboratory (RAL) of the Science and Technology Facilities Council (STFC). The NSCCS Cluster is called Slater. Slater is a Silicon Graphics Altix UV 2000 with 512-cores and has a memory of 4TB with 22TB of scratch work space. CPUs: 64 x Intel E5-4620 v2 2.6GHz 8 core Ivybridge CPUs. SUSE LINUX Enterprise 11 is installed on Slater. Users familiar with other flavours of Unix should find no difficulty in using the machine.

All runscripts for each of the software packages are located in the `$CHEM` directory. Users are advised to look at the relevant man pages before submitting their jobs. The documentation relating to running jobs on the machines is located in `$CHEM` on Slater (see section 6).

2.2 How to Log In

Users can only connect to the machine using the Secure Shell Client (ssh2). Detailed information on how to start SSH on different machine architectures is given below. SSH is a program that can be used to log into another computer over a network, to execute commands on a remote machine,

and to move files from one machine to another. It provides strong authentication and secure communications over unsecure channels. It is intended as a replacement for rlogin, rsh, and rcp. Additionally, SSH provides secure X connections and secure forwarding of arbitrary TCP connections. The SSH client is available on most Linux/Unix and Mac OSX machines. For Windows PCs, there are many SSH clients available in the form of freeware and commercial versions. For further information on SSH see: http://en.wikipedia.org/wiki/Secure_Shell

Connecting to Slater from Linux/Unix machines

If you are using a Unix or Linux machine, it generally comes with SSH and will either be automatically installed or available via your package management facility. If SSH is not already installed on your machine, please ask your local Linux/Unix administrator for advice.

To connect to Slater:

1. Open a terminal window.
2. Type the following at the prompt:

```
ssh -l userid slater.rl.ac.uk
```

where `userid` is your RAL userid. You will now be prompted for your password.

Connecting to Slater from Mac OSX machines

SSH should already be installed with Mac OSX as part of the Terminal application.

To connect to Slater:

1. Open *Finder*, then open *Macintosh HD* ⇒ *Applications* ⇒ *Utilities*. Open *Terminal*.
2. At the terminal, type the following at the prompt:

```
ssh -l userid slater.rl.ac.uk
```

where `userid` is your RAL userid. You will now be prompted for your password.

Connecting to Slater from a Windows PC (Windows 7)

Windows users can use either PuTTY (http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html) or MobaSSH (http://mobassh.mobatek.net/) which are free of charge.

e.g. To connect to Slater using PuTTY (*latest release version (beta 0.62)*) on Windows 7.

1. Start *PuTTY*.
2. A *PuTTY Configuration* window will appear.
3. Enter `slater.rl.ac.uk` into the *Host Name* box. Select *SSH* as the connection type. Click *Open*.
4. A window will be opened and prompt for your login name. Enter your RAL userid and press enter.
5. You will now be prompted for your password. Type in your password and press enter to log in to the machine.

2.3 How to Access X-Windows Applications (including Graphical Packages)

To use any of the graphical interfaces on Slater, some kind of X-Windows emulator is required and you will need to log in to the machine using SSH X11 Tunnelling (X11 Forwarding). The same is true for all other X-Windows applications you wish to access remotely.

From Linux/Unix

To set up a Linux/Unix machine to use SSH X11 Tunnelling, you need to add Slater to set of allowed hosts and set the DISPLAY environment variable. This can be done automatically using the following command:

```
ssh -X -l userid slater.rl.ac.uk
```

where `userid` is your RAL userid. You will now be prompted for your password to log in to the machine.

Alternatively, you may set up everything manually in the following way:

1. Open an xterm terminal.
2. Type the following to add Slater to the list of host names allowed to make connections to the X server:

   ```
   xhost +slater.rl.ac.uk
   ```
3. ssh to Slater following the steps as shown in section 2.2.
4. You now need to set the DISPLAY environment variable for the X-server to display the graphical interface on the local machine.

   If a user is using csh/tcsh shell on Slater, use the following command:

   ```
   setenv DISPLAY display-machine-IP:0.0
   ```

   If a user is using sh/ksh/bash shell on Slater, use the following command:

   ```
   export DISPLAY=display-machine-IP:0.0
   ```

   where display-machine-IP is the IP address of the machine you wish the display to appear on.

From Mac OSX (e.g. 10.6.8)

Open the *X11* application from *Utilities* and use the following command:

```
ssh -X -l userid slater.rl.ac.uk
```

where `userid` is your RAL userid.

You will now be prompted for your password.

From Windows PC (Windows 7) using Xming with *PuTTY*

On Windows machines, users will need to use an X-Windows emulator. This example uses Xming (http://www.straightrunning.com/XmingNotes/). This example uses the public domain release Xming-mesa Version 6.9.0.31.

1. Start Xming. After clicking on it, Xming is launched automatically and will be running in the background.
2. Start *PuTTY*.
3. A *PuTTY Configuration* window will appear. You will be given the option to put in the hostname where you wish to connect to. But before you connect to Slater, you will need to change one of the options.
4. Select *Connection* → *SSH* → *X11* from the Category. Check the box to enable X11 Forwarding.
5. Select *Session* from the Category.
6. Enter slater.rl.ac.uk into the *Host Name* box. Select *SSH* as the connection type. Click *Open*.
7. A window will be opened and prompt for your login name. Enter your RAL userid and press enter.
8. You will now be prompted for your password. Type in your password and press enter to log in to the machine.

9. An X-Windows window will automatically open whenever an X-Windows program is started in the remote Unix host.

An alternative open source X-Window System for Microsoft Windows is available via the use of Cygwin/X. Cygwin/X is a port of the X-Window System to the Microsoft Windows family of operating systems. Cygwin/X is installed via Cygwin's setup.exe and the installation process is documented in the Cygwin/X User's Guide. Cygwin/X can be downloaded at:

http://x.cygwin.com/

Note: Please note that if the graphical package requires OpenGL (e.g. GaussView), you will need to use Exceed 3D if you are using Hummingbird Exceed, or if you are using Cygwin/X, you should download the OpenGL library files during installation.

3 General notes on machines

3.1 Login Shell

The login shell is the command line interpreter that the system starts for you when you first log in so that you can execute commands. The login shells supported by Slater are the standard Bourne shell (sh), Korn shell (ksh), the C shell (csh), the extended (or "turbo") C shell (tcsh), and the Bourne again shell (bash). The default shell on Slater is the bash shell.

3.2 Shell Environment File

When you log in, various default configuration files are executed which set up the default environment. After the default configuration has been set up, your personal environment is configured using the relevant shell environment file in your home directory. These are listed below for each shell type.

`sh`	`.profile`
`csh`	`.cshrc` and then `.login`
`ksh`	`.profile`
`tcsh`	`.cshrc` and then `.login`
`bash`	`.bash_profile` or `.bash_login` or `.bashrc` or `.profile`

When your account was created you will have been given a standard version of the relevant file(s) for your login shell. Different files may be executed when a shell is started that is not a login shell, and also when a shell exits. More information can be found in the Unix man page for the shell you are using. For example, to view the man page for `bash`, type the following at the Unix prompt.

```
man bash
```

3.3 Changing your Shell

When your account is set up you will be allocated the default shell bash shell as your login shell. You can check to see which shell you are currently using by typing the following command at the Unix prompt:

```
echo $SHELL
```

To change this to another supported login shell, you can use the command `chsh`. The new login shell must be one of the approved shells listed in the `/etc/shells` file unless you have superuser privileges. Note that when changing a shell, the full path to the new shell must be given (e.g. `/bin/ksh`, `/bin/csh`, `/bin/tcsh`, `/bin/bash`).

For example, if you type:

```
chsh
```

at the Unix prompt, then you should see the following:

```
Old shell: /bin/bash
New shell:
```

The old shell listed is the one currently running (bash) and this can be left unchanged by pressing Enter. Alternatively to change shells, enter the full pathname of the shell you wish to use. For example, to change to tcsh, enter:

```
New shell: /bin/tcsh
```

The change to your shell will generally take effect the next time you log in.

More information on Unix shells may be found at:

http://www.faqs.org/faqs/unix-faq/shell/shell-differences

4 Files and Filestores

4.1 Home Directories

The home file store (home directory) is the most important of all file systems. This is where the system places you when you initially log in. For NSCCS users, the default home file store is located at:

```
/home/slater/userid/
```

where `userid` is your login name (you can always check to see which directory you are currently in by using the `pwd` command).

The home directory is regularly backed up but it is of a limited size (see section 4.3 below). Users are advised to copy files back to their local machines on a regular basis and not to use their home directories on Slater for permanent storage (see section 4.4).

4.2 Use of Temporary File Systems

Temporary files should be on the `/scratch` file systems and should be used by batch jobs for all work files used during a run. `/scratch` provides a cheap resource for storing files that may be required over multiple batch jobs. Files on `/tmp` or `/scratch` not belonging to executing jobs may be deleted without notice in order to make room for the large temporary disk storage that is essential to many users.

When using the runscripts provided for the chemistry software packages on Slater, large work files will automatically be written to these file systems and all relevant output files copied back to the directory from where a job is launched. Sometimes additional files may be needed by the user, e.g. to restart a job. If these are created on `/scratch`, the user should make sure that the files are copied back to their home directory as soon as their job has finished to avoid them being deleted when the file systems are purged.

Users are advised not to use `/tmp` or `/scratch` as extra file space if their allocations elsewhere run out! If users require extra file space, they should contact NSCCS Support by email (nsccs.support@stfc.ac.uk).

4.3 File System Controls

We do not have 'hierarchical storage management' software for Slater. The advantage of this is that your files are always available without having to wait for recall from tape, the disadvantage is that we have to apply controls to stop users abusing the system.

When you are first registered on Slater you are allocated a 'soft' limit on storage that you can exceed for up to 14 days before the system prevents you from creating further files.

When you hit the limit you can clean up unwanted files as necessary and/or request a larger file allocation. If you request a significantly larger allocation, and can justify it, for instance by referring back to your original application, then a 'hard' limit will be set which will prevent you creating further files as soon as you reach it. Users with large file store allocations should manage their files so that this does not happen too often!

4.4 Data Transfer to and from Slater

There are two ways to transfer data to/from the machines:

• scp (secure copy)

• sftp (secure file transfer protocol)

From Linux/Unix

Users can simply use the commands scp or sftp to transfer data.

e.g.

```
sftp userid@slater.rl.ac.uk
scp filename userid@slater.rl.ac.uk:target_directory
```

You will be prompted to enter your password.

For more information, please refer to the corresponding Unix man pages.

From Max OSX

Users can use the same commands as above via the *Terminal* application.

Alternatively, there are many open source software application such as CyberDuck (http://cyberduck.ch), which is a FTP/SFTP Browser, where users can log in via the interface to copy files to/from the machines.

From Windows PC

There are several free applications that can be used to transfer files. One example is the free SFTP/SCP client for windows called WINSCP (http://winscp.net).

4.5 How to Recover Files if Deleted Accidentally?

If the files you would like to recover are deleted in the last week, users can retrieve them from their `snapshot` directory.

You need to return to your home directory by typing:

```
cd ~
```

Then you can change into the `snapshot` directory:

```
cd .snapshot
```

In this directory you will find sub-directories for each of the last 7 days, including today so you

could restore any files deleted in the last 7 days from the `.snapshot` directory for that day.

Please note files can only be recovered if there has been a backup overnight.

For files deleted over a week, users should contact NSCCS Support by email (nsccs.support@stfc.ac.uk) to recover the files from backup tapes. Normally files up to two weeks old may be restored.

5 Editing

5.1 Available Editors

The main text editors on Slater are `vi`, `emacs` and `nano` (a GNU clone of `pico`) which are all terminal based. There are other editors such as `xemacs` and `nedit` which require the use of X-windows. Please refer to the corresponding Unix man pages for details on how to use the editors.

6 Software

We provide a wide range of software packages on our machines, applicable to research across all fields of chemistry. More detailed information on the software packages we support can be found at: http://www.nsccs.ac.uk/software.php

If there is a software package that you would like to use on our machines but it is not currently implemented, please contact the Service Manager Dr Helen Tsui by email (helen.tsui@imperial.ac.uk). Please note that users may not run their own "home-grown" software packages on Slater unless they are willing to donate these packages to the NSCCS and make them generally available to all users. The exceptions are non-CPU intensive pre- and post-processing scripts which may be used at the discretion of the Service Manager.

6.1 Running Jobs

Runscripts (e.g. `runadf2013`, `rung09_d01`) are available for all the chemistry software packages on Slater. These are installed in the directory `$CHEM` on Slater. Runscripts are shell scripts written for executing each software package. Each runscript has a man page and users are strongly advised to read this before running jobs. The man pages can be viewed by typing `man` followed by the name of the runscript. For example, to view the man page for Gaussian 09 Rev.D.01, type the following at the Unix prompt:

```
man rung09_d01
```

Users should always use these runscripts to ensure that the relevant environment variables and paths are set correctly. They also help the NSCCS to keep track of where CPU time is being used on the machine. The CPU time deduction from users' accounts is not related to these runscripts but is done automatically by the Unix accounting system, so users will gain nothing by running their jobs without using them.

A full list of runscripts can be found on the NSCCS web site:

http://www.nsccs.ac.uk/ug_runscripts_slater.php

6.2 Submitting Jobs

All jobs should be run through the LSF batch queuing system (see section 7), unless they require very little in the way of resources (both in terms of memory and CPU time). Users should be aware

that memory limits and CPU limits apply to interactive work and their jobs will be killed automatically if they exceed these.

7 Batch Jobs

7.1 Structure of the Queuing System

Batch jobs are submitted via the queuing system. There is a selection of queues available with different configurations. Please read the man page for the software package you wish to use before submission. For a full list of software packages available on Slater, please visit this web link for details:

http://www.nsccs.ac.uk/software_list.php

Specific information about a particular queue can be obtained by using the command:

```
bqueues -l <queuename>
```

Alternatively information about all the queues can be obtained by using the command:

```
bqueues -l
```

7.2 Queues

The configuration of the batch queues for running work on Slater is listed below. Each value given is the limit of the resource in that queue.

Queue name	Priority	CPU Time Limit (min)	Wallclock Time Limit (min)	Memory Limit (MB)	Number of processors	Maximum number of processors per user	Maximum number of jobs per queue
`a1`	15	60	180	16777216	1 - 4	12	32
`a2`	10	3600	7200	16777216	1 - 16	32	160
`a3`	5	15000	18000	235929600	1 - 64	64	192
`a4`	4	90000	18000	235929600	8 - 64	64	192
`R`	*10*	*120000*	*180000*	*235929600*	*1-512*	*512*	*512*

The *R* queue is the restricted queue reserved for use by NSCCS staff only.

7.3 Working in Batch

7.3.1 Introduction

The batch job control system Slater is the Load Sharing Facility (LSF) from Platform Computing Corporation. This provides a set of batch queues to which users can submit batch jobs. The LSF system then manages the running of the batch work selecting jobs from the different queues depending on the relative priorities of the batch queues and available resources for running batch work. LSF is similar in concept to NQS or PBS and users familiar with these systems will find little difficulty in converting to using LSF. The command used to submit jobs to LSF is `bsub`.

The batch job control is based around a job script that contains the instructions to run the job and some optional control parameters. At the simplest level the job script is submitted and controlled with three commands:

`bsub` to submit a batch job

`bjobs` to check on the status of batch jobs

`bkill` to cancel a batch job and prevent execution

All batch commands listed in this guide have detailed Unix man pages which provide full details of command usage.

7.3.2 Fairshare scheduling

The queuing system on Slater utilises fairshare scheduling. This scheduling divides the processing power of the LSF cluster among users and groups to provide fair access to resources. By default, LSF considers jobs for dispatch in the same order as they appear in the queue (which is not necessarily the order in which they are submitted to the queue). This is called first-come, first-served scheduling. The fairshare scheduling prevents a single user monopolising the cluster's resources for a long period of time. The fairshare scheduling used on Slater is based on the resources (CPU time) that the users have consumed in their jobs. When fairshare scheduling is used, LSF tries to place the first job in the queue that belongs to the user with the highest dynamic priority.

7.3.3 Batch Job Scripts and Job Submission

Each batch job should have a control script which contains the instructions necessary to perform each part of the job in turn. The instructions can be anything that you would normally type from the Unix command line to perform the tasks interactively.

You must give LSF options to inform it about the needs of your job. Some of the basic options are described below.

-n This is used to request the number of CPUs.

-W This is used to request the wall clock time used. This means that your job will automatically finish after that amount of time is used up if it has not already finished. Measured and specified in minutes.

-c The -c option is similar to -W in that it is a way of restricting the amount of time your job runs for. However -c is the total amount of CPU time used. Measured and specified in minutes.

-q This is used to specify which queue your job runs on.

-J This is to give your job a name which can be useful to identify which of your jobs are running when using some of the LSF monitoring .

-e This is to specify the name of the file where the stderr should be outputted to.

-o This is to specify the name of the file where the stdout should be outputted to. If only the -o option is specified, then the stdout and stderr are merged into the specified file.

-R This is to specify the resource requirement for a particular job.

There are two ways to specify the LSF job submission options. The first is by giving the options on the 'command line'. For example, a simple script (`jobscript`) to run a Gaussian calculation might contain the line:

```
$CHEM/rung09_d01 < file.inp > file.out
```

where `$CHEM/rung09_d01` is the runscript for executing the software package, `file.inp` is the Gaussian input file with the results to be written to `file.out`.

Then all that is needed to submit the job is:

1. To make sure the script has execute permission by typing:

```
chmod u+x jobscript
```

2. To submit the job by typing a `bsub` command, e.g.

```
bsub -n 4 -J my_job -q a1 -o output jobscript
```

This will run a Gaussian job on 4 processors, writing the stdout to a file called `output` with the job name `my_job`.

Alternatively, the LSF job submission options can be placed in the submission script written in a format which makes them look like comments in a Unix shell. The LSF syntax for submission options is:

```
#BSUB <option> <value>
```

Any of the command line options to the `bsub` command can be specified. A script with embedded commands would therefore be similar to:

```
#BSUB -n 4
#BSUB -J my_job
#BSUB -q a1
#BSUB -o output
$CHEM/rung09_d01 < file.inp > file.out
```

Note that there is one difference in the way that this script must be submitted in order for LSF to read the embedded options. The `bsub` command only interprets embedded options if the script is supplied as the stdin of its command line. This means that the script must be submitted as follows:

```
bsub < jobscript
```

If the script is just specified on the command line then the embedded options are ignored.

Please note if the redirection sign (`<`) is missing in the above command, then the job will be submitted to the `a1` queue by default even if you have specified a different queue name in your `jobscript`.

It is also possible to put the input file for the software inside a submission script. If this method of submission is selected, the output file will not appear in the directory where the job is submitted until the job has completed. While the job is still running, users can access the temporary output file in the following directory.

```
/home/slater/userid/.lsbatch
```

Users can also use the command `bpeek jobid` to tail the output while it is running, where `jobid` is the numeric identifier given to the job by LSF.

e.g.

```
bpeek 12345
```

Below is an example of a Gaussian input file placed inside a submission script.

```
#BSUB -n 4
#BSUB -J my_job
#BSUB -q a2
#BSUB -o output
$CHEM/rung09_d01 << EOF
%nproc=4
%chk=water
```

```
# b3lyp/6-31G* opt

Water - B3LYP geometry optimisation

0 1
O
H 1 0.96
H 1 0.96 2 109.471221

EOF
```

7.3.4 Checking Job Status

The command to check the status of LSF jobs is `bjobs`. On its own, `bjobs` will return a list of all your jobs and whether they are queued or executing. Useful options are:

`bjobs -u all`	to see the jobs of all users
`bjobs -q queue_name`	to restrict the output to a single queue
`bjobs -l jobid`	to see more detailed information about a particular job

where `jobid` is the numeric identifier given to the job by LSF and is displayed as one of the fields in the `bjobs` command.

Users can also check the status of the queues by using the command `qstat -a`, which displays information such as how many jobs are currently running and pending on the queues.

7.3.5 Deleting Jobs from the Job Queue

The command to remove a queued job from LSF is `bkill` and the syntax is:

```
bkill jobid
```

where `jobid` is the numeric identifier given to the job by LSF.

You can only cancel jobs that you have submitted yourself. The job should be removed from the queue after a short while. If the job still remains on the queue, users should try using the following command to kill the job:

```
bkill -s KILL jobid
```

If this fails, users should contact NSCCS Support by email (nsccs.support@stfc.ac.uk).

7.3.6 Advice on Using Batch

Please try to keep some check on the physical memory size used by batch jobs. If a job does not require large physical memory then please do not submit jobs to the large memory queues as this will block the running of jobs that do require large physical memory. It will probably also result in a longer turn around time for your job.

The physical memory size is the resource being requested as a requirement when a batch job is submitted with a memory limit specification. Alternatively the memory limit may come from submitting a job to a specific batch queue. The amount of memory being used by a running job is

one of the statistics reported by the `bjobs -l` command. Look for the section of output which looks like:

```
Fri May  1 10:30:08: Resource usage collected.
                     The CPU time used is 114451 seconds.
                     MEM: 368 Mbytes;  SWAP: 475 Mbytes
                     PGID: 301;  PIDs: 301 349 31895
```

This shows that the job is currently using 368 Mbytes of memory.

7.3.7 Output File Selection

By default the output from the LSF batch job will be returned as an email message. To have the output directed to a file, the "`-o`" and "`-e`" options should be used. e.g.

```
bsub -o output.log -e error.log jobscript
```

This will direct the messages sent to `stdout` to the file `output.log` and the messages sent to `stderr` to the file `error.log`.

7.3.8 Queue Selection

Different jobs have differing CPU and memory requirements. For this reason the different queues listed in the section 7.2 are available to users. If no CPU and memory requirements are specified then `bsub` will place the job in the queue `e1` by default. Selecting another queue for a job can be accomplished by either submitting the job directly to a particular queue or by specifying resource requirements for the job. For example:

```
bsub -q a1 jobscript
```

will submit the job to be run in the `a1` queue. The job will then run with the CPU and memory limits of the `a1` queue.

Alternatively:

```
bsub -c 3:30 -M 358400 jobscript
```

will specify that 3 hours 30 mins of CPU time and 358400Kb of memory are required. LSF will then choose the most appropriate batch queue into which to place the job. When the job runs it will have a CPU and memory limit as specified when the job was submitted.

7.3.9 Chained Batch Jobs

Batch jobs can be chained together to run one after the other using the "job dependency" options of the `bsub` command. e.g.

```
bsub -J Mysub1 -q queuename jobscript1
bsub -J Mysub2 -q queuename -w 'ended(Mysub1)' jobscript2
```

The first command submits the script `jobscript1` to be run in batch with a jobname of `Mysub1`. The second command submits jobscript2 to be run but is dependent on the `Mysub1` batch job to have ended before it can start. In the same manner you could add

```
bsub -J Mysub3 -w 'ended(Mysub2)' jobscript3
```

etc.

7.3.10 NQS Compatibility

For those familiar with NQS, LSF provides some support for public domain NQS style commands. NQS users will however need to move to using the native LSF interface for the extra functionality that this interface provides.

7.4 Cluster wide commands

On Slater, the following command can be used to see which processes are running on which processors:

```
jobinfo userid
```

7.5 Further Information

The current status of CPU usage and queue information is available from this web link:
http://sct.esc.rl.ac.uk/NSCCS/status.html

There is also some information in the Unix man pages on the LSF batch system, type the following at the Unix prompt:

```
man lsfintro
```

and

```
man lsfbatch
```

If you have any other queries about LSF then please contact NSCCS Support by email (nsccs.support@stfc.ac.uk).

8 The NSCCS Web Portal

The NSCCS has a web portal interface where users can access our machine Slater via a web browser.

The portal interface offers the following:

- Simple Login
- Access to Home Directory
- Simple Job Submission Forms - Application Specific
- Simple Monitoring Interface
- Serial and MPI Jobs
- Editable Job Submission Forms
- Continuous View of Job Output
- Application Templates Set Up for Gaussian 09 Rev.B.01, Gaussian 09 Rev.C.01, Gaussian 09 Rev.D.01 and Gaussian utilities.

For more details on how to access the web portal, please visit the web link: http://www.nsccs.ac.uk/portal.php

9 Running Jobs on NSCCS Machines

9.1 Running Jobs in Parallel

Parallel computing is the simultaneous use of multiple compute resources to solve a computational problem. The parallel environments allow individual programs to distribute their workload across a number of CPUs to undertake parallel computation, resulting in a reduced wall clock time for a job. Many of the software packages on Slater can be run in parallel. Users should check the relevant man page for more information.

9.2 Memory Allocation

Users should be aware that for parallel computing, there are different memory architectures available. The choice of memory allocation depends on how the software package is parallellised. There are two main architectures, one is shared memory and the other is distributed memory. Slater uses shared memory.

9.2.1 Shared Memory

Shared memory is where all processors on a computer have direct access to the common physical memory such that the parallel tasks of a job will all have access to share the physical memory available on the hardware. Generally speaking, the memory allocated in an input file in this case corresponds to the total memory allocated for the job.

9.2.2 Distributed Memory

Distributed memory is physical memory that is not common to all processors. In this case, it is necessary to use some kind of communication to access memory on other machines where other tasks are executing. There are many parallel programming models that can provide the communications such as MPI, SHMEM and Linda. Generally speaking, the amount of memory specified in an input file in this case corresponds to the memory allocated on each processor.

9.2.3 MPI

The Message Passing Interface (MPI) is a communication protocol that offers Application Programming Interfaces (APIs). It allows computation to be distributed across multiple CPUs on different machines. The communication itself is a two-step process. For example if there are two processors A and B, processor A will make a call to send the data and processor B will make a call to receive the data. The two processors must cooperate with each other where processor B must make a library call to accept the data before using it. An example of a software package on Slater that uses MPI is ADF.

9.2.4 SHMEM

SHMEM refers to the shared memory access library available on Cray, SGI machines and HP Alphaserver SC (and others). The SHMEM library provides the capability to have a processor read and write the memory of another processor without that processor's cooperation. This is called active messaging. For example, processor A can read data from processor B without processor B's participation and it does not interrupt processor B's CPU. The SHMEM routines minimise the overhead that is associated to the communication between the processors. Hence it has a lower latency and higher bandwidth than MPI. An example of a software package on Slater that uses SHMEM is Molpro.

9.2.5 TCP Linda

TCP (Transmission Control Protocol) Linda is a parallel execution environment which has been used to create a parallel version of the software package Gaussian for local area network and some distributed memory multiprocessor environments. The Linda parallel programming model involves a master process, which runs on the current processor, and a number of worker processes which can run on other nodes of the network.

10 Monitoring your Resources

10.1 Accounting on NSCCS machines

Your grant on the NSCCS machines is allocated as a number of CPU hours and an end date, based on the amount of resources awarded in your application. The default account for CPU charging is displayed immediately after logging in to Slater. You can find out how your usage is progressing via the `acct` command which has several options summarised below:

`acct qcomb user userid`	lists accounts userid can use
`acct qcomb acct chemxxx`	lists users who can use sub-proj
`acct qusage chemxxx`	reports usage of sub-proj
`acct help`	lists other available acct commands

To find out the local subproject(s) allocated to your grant, type:

```
acct qcomb user userid
```

To check on your current usage and allocations, type:

```
acct qusage chemxxx
```

where `xxx` should be replaced by the number of your subproject.

e.g.

```
acct qusage chem123
```

Please note that the account information is only updated overnight, so the amount of CPU used during the current day will not appear until the next day.

Users should be aware that the CPU times reported in the output files and the batchout files may be incorrect for some parallel applications. Although there is currently no way to accurately record the CPU time used by some of these jobs, the correct amount of time will be charged for. Therefore users are advised to monitor their CPU time usage carefully using the `acct qusage` command.

The most accurate method to check how much CPU time has been used by an individual user is to use the following command which gives a break down of all projects the user running the command is on.

```
acct qusercpu [BOTH | MAGELLAN2 | SLATER] <start-date> <end-date>
```

e.g.

```
acct qusercpu  BOTH 01-Jan-2007 01-Jan-2012
```

Users can also view the following file to find out the time used for each of their jobs in the current month but time may not be accounted for in certain parallel jobs.

```
/var/log/chemuse.log
```

An example of what is printed in the `chemuse.log` file is given below.

```
Jun  9 16:34:18 slater ht3: /usr/local/Chem-Apps/Gaussian/g09_d01/g09
et=13.509 ut=42.175 st=5.976 mrKb=8491064 adMb=0 asMb=0 bi=248592
bo=22176 LSF=221.a3 ncpu=4
```

The date and time of the job completed, the machine the job was running on (e.g. `slater`), the userid (e.g. `ht3`), the location of the program's executable (e.g. Gaussian 09) are given. The CPU time for the job is the sum of the usage time (`ut`) and the system time (`st`) (e.g. CPU=42.175(`ut`) + 5.976(`st`)=48.151). All times are reported in seconds. Each of the jobs is identified by the LSF batch request ID number (e.g. `211`) and the name of the queue the job was submitted to (e.g. `a3`).

Older chemuse.log files are available in the `/usr/local/locallog/chemusage` directory.

10.2 Groups and Grants

Each grant of time on the NSCCS machines is allocated a Unix group which will be equivalent to the subproject used by the ACCT system (type `acct help` for further information on the various commands which can be used to find out the status of your grant). Most users are registered with only one project so are in only one Unix group and for them there is nothing further to worry about.

For those users who are registered to use more than one group some thought must be taken about which group it is appropriate for activities to be charged to. To find out the default account, type the following at the Unix prompt:

```
id -g -n
```

If you would like your default group changed, please contact NSCCS Support by email (nsccs.support@stfc.ac.uk).

10.3 Interactive Work

If you wish interactive work to be accounted to an alternative group, type the following at the Unix prompt.

```
newgrp chemxxx
```

where `chemxxx` is the alternative group name. Interactive processes and any processes you fire up as background work will then be accounted to group `chemxxx`. The change will stay in effect until you log out. If you wish to return to your default group, type `newgrp` at the Unix prompt.

e.g.

```
newgrp chem123
```

10.4 Batch Work

If you wish batch work to be accounted to a group other than your default group then type:

```
bsub rungroup groupname batch_script_file [script arguments]
```

e.g.

```
bsub -q a1 rungroup chem123 jobscript
```

Your job will appear in the queuing system with the name "`rungroup groupname`" where `groupname` is the group the batch work will be accounted to. Please be aware that the Unix group is used principally to control file access; some experimentation may be required to achieve any file-sharing across projects which you need.

10.5 At the end of a Grant

When either your grant has reached its end date or all the allocated time has been used, if you are working on only one project you will be disabled. If you are working on more than one project your

`userid` will no longer be able to `newgrp` to the Unix group corresponding to the terminated project.

If users need to retrieve files from their accounts after they have been disabled, they should contact NSCCS Support by email (nsccs.support@stfc.ac.uk). Accounts that have exceeded their expiry dates may be extended at the discretion of the NSCCS.

10.6 Disk Quota

Users can monitor their disk quota by typing the command `panfs_quota` at the Unix prompt. If your disk usage is reaching your limit limit, a warning message will appeared on screen immediately after you log in.

11 Documentation

Information on all commands on the system is available using the standard Unix 'man' tool. Unix-style man pages are available for all the runscripts provided for the chemistry software packages. Documentation for the software packages can be found in the directory `$CHEM/doc` on Slater.

More information on how to use some of the software packages can be found on the NSCCS website under:

http://www.nsccs.ac.uk/user_softintro.php

Step-by-step user guides to address some of the more common problems users have can be found at:

http://www.nsccs.ac.uk/user_guides.php

12 Keeping Up to Date

12.1 NSCCS News

NSCCS news can be found at the following web link:

http://www.nsccs.ac.uk/news.php

The news is updated at regular intervals, and occasionally messages are placed here if there are particular problems.

There is also hardware news for Slater, regularly updated by the support staff at STFC RAL and can be found at the following web link:

http://www.nsccs.ac.uk/news_slater.php

12.2 Scheduled Maintenance and Updates

The machines may be unavailable during periods of scheduled maintenance and system updates. Users will be notified in advance of these sessions via email, on the system itself and via the news page on the NSCCS web site.

12.3 News and the NSCCS Mailing List

When you log in you will see a list of unread news items relating to system matters. Read an item by typing `news filename` at the Unix prompt. These news items may also be put on the news page on the web, along with more general NSCCS information.

Users may also sign up to the NSCCS RSS news feed at the following web link:
http://rss.esc.rl.ac.uk/NSCCS.xml

Users will automatically be added to one of our JISC mailing lists when they are registered (PIs to UK-CCF, other group members to NSCCS-USERS). Important service information will be posted on the NSCCS web pages (www.nsccs.ac.uk) and circulated via these mailing lists. Please ensure that you inform us if you change your contact details.

12.4 Support

Software & Hardware

All hardware and software related queries should be sent to NSCCS Support
(email: nsccs.support@stfc.ac.uk).

Training

All queries relating to software training, please contact Dr Alexandra Simperler (email: a.simperler@imperial.ac.uk).

Login Issues and Password Reset

All login related issues (e.g. SSH access problems and password reset) should be sent to NSCCS Support (email: nsccs.support@stfc.ac.uk).

Other Queries

For all other service queries, please contact the Service Manager Dr Helen Tsui by email (email: helen.tsui@imperial.ac.uk).